A Survey of Self-Organization in Wireless Networks

Kevin L. Mills
National Institute of Standards and Technology
kmills@nist.gov

Current and future uses suggest that wireless communications and networks should become an essential, possibly predominant, part of the world's global information infrastructure. Do you own a cell phone? If not, do you own a cordless phone? How about a laptop computer with one or more built-in wireless interfaces, such as WI-FI (wireless fidelity) or Bluetooth? Or maybe you have a personal digital assistance with a wireless capability? Perhaps you have already used some of these devices to access the Internet while flying in a commercial jet. Do you use geo-location devices based on the global positioning system? Are you looking ahead to the day when multimedia devices in your home entertainment system transmit sound and movies around your house without wires? How about future highway systems where cars wirelessly receive information relevant to their surroundings? Or even where cars wirelessly exchange heading and velocity information in order to improve traffic flows and to avoid collisions?

Beyond these well-known or easily imagined uses of wireless technology, ongoing research in industry and academe promises to open a range of new applications. For example, advances in mobile ad-hoc networks (MANETs) promise easily deployable local and metropolitan networks without running communication cables to connect nodes. Technology of this type seems aptly suited for deployment when responding to natural disasters or terrorist attacks that might destroy or disable preexisting wired communications infrastructures. As another example, scientists and engineers envision deploying tiny, wireless sensor and compute nodes (sometimes called “motes”) that can form a network to make and convey measurements from small physical areas and to aggregate a picture of the situation across a geographic expanse. Many applications are imagined: measuring ocean temperatures and currents, analyzing moisture content in soils, gauging ground motions, assessing sunlight in forests, and monitoring stresses in structural supports of large buildings. Further, tomorrow’s hospitals will likely find patients outfitted with medical devices that use wireless communication to transmit vital signs for monitoring and analysis. Even in something as seemingly mundane as grocery shopping, engineers envision a future where product packages include wireless, radio-frequency identification (RFID) tags that could enable quick, automated checkouts and timely reordering and restocking.

What might this growing array of applications mean for the way we design, deploy, and manage wireless networks? Several things should be abundantly clear. The number of networked devices will become quite large, maybe exceeding thousands per person. Further, wireless communication implies device portability and mobility at a variety of speeds. Potential exists for wireless networks to vacillate between sparse and dense connectivity as the population of reachable devices varies. Thus the number of devices, communications channels, and data transmissions will become too large, varying, and uncertain to be deployed and managed with the costly, labor-intensive techniques in use today. Instead, wireless devices and networks must become adept at *self-organization* – allowing devices to reconnoiter their surroundings, cooperate to form suitable network topologies, and monitor and adapt to changes in the environment, all without direct human intervention.

This paper surveys research investigating how self-organization might be applied in tomorrow's wireless networks – to share and manage resources, to form and maintain structures, and to shape behavior. The paper begins, in the next section, by considering two aspects of self-organization: (1) self-organization as a natural phenomenon that might arise in any distributed system and (2) self-organization as a design principle that can be applied to design and engineer distributed systems. These two aspects represent a duality (and tension) regarding self-organization. On the one hand, self-organization is likely to appear in any distributed system, where numerous components interacting on a microscopic level lead to a range of macroscopic behaviors that emerge, or self-organize, at a global level. In this view, self-organization is a natural consequence of distributed systems; however, the behaviors that emerge are not controlled and thus may be unpredicted and undesired. On the other hand, system components may be endowed with specific rules that lead to the emergence of intended and desired global behaviors. In this view, self-organization is a design principle employed to achieve specified objectives. The paper surveys a number of models that could serve as underlying design principles for self-organization in distributed systems.

In a third section, the paper examines specific attempts to apply self-organization in wireless networks for particular aims, divided into five categories: (1) resource sharing (e.g., of spectrum, bandwidth, and processing capacity), (2) structure formation and maintenance (e.g., of topologies, software components, and conversational syntaxes), (3) behavior shaping (e.g., of routing, information dissemination, querying, and task or service placement), (4) resource management (e.g., to synchronize time and conserve power), and (5) resiliency (e.g., repairing faults and resisting attacks).

In a discussion section, the paper assesses the current state of the art in self-organization in wireless networks and ponders some open questions. One question centers around self-organization to achieve competing design objectives. Can self-organizing design models provide suitable solutions to such problems? Another question arises from the tension between natural and intentional self-organization. Can intentional self-organization be affected negatively by naturally occurring self-organization in a distributed system? Alternatively, can multiple self-organizing behaviors, whether designed or unintended, interfere or interact with each other to form unintended consequences? In other words, what must designers surrender in order to exploit self-organization?

Self-Organization

"Sometimes a system with many simple components will exhibit a behavior of the whole that seems more organized [or ordered] than the behavior of the individual parts…[such] complex phenomena are called emergent behaviors [or properties] of the system."[1] Emergent properties seem to arise naturally in complex adaptive systems "via a process of self-organization, autocatalysis, or autopoiesis."[2] Self-organization appears in many natural and man-made systems, such as biological organisms[3], ecosystems[4], food webs[5], geological systems[6], metabolic networks[7], transportation networks[8], communication networks[9], and stock markets[10]. The following subsection considers self-organization as a natural phenomenon, providing a brief summary of current scientific thinking on how self-organization arises, what advantages might accrue from self-organization, and how self-organization might be detected or measured. A subsequent subsection considers self-organization as a design principle and discusses several models that might be applied to achieve intentional self-organization.

Self-Organization as a Natural Phenomenon

Complex adaptive systems consist of "mutually entangled" components, where change in any component propagates to many (or all) other components through component interactions that exhibit a ripple effect over space and time. "Complex systems will typically exhibit a tangle of interconnected positive and negative feedback loops, where the effects of any change in a component cascade through an increasing number of connected components."[11] As a result of such interactions, system (or global) state tends to move over time toward some coherent pattern. This is the essence of self-organization. Because self-organized patterns arise from many interactions spread over space and time, it can be difficult to predict what system state will appear. Some system states are said to be emergent properties because they have no meaning for individual components. For example, gas (a collection of molecules) exhibits both temperature and pressure, which might be seen as measures of the strength of interactions among molecules.[11]

Given that self-organization entails movement from less ordered system state toward a more coherent pattern, what patterns might be observed? One possibility is equilibrium, where system state reaches some fixed point. A second possibility is oscillation, where system state cycles repeatedly through the same series of points. A third possibility is chaos, where system state wanders forever through a non-repeating set of points. Oscillation and chaos may be considered terminal states, while equilibrium may be transient. Some scientists have noted a tendency for equilibrium states in certain systems to exhibit a delicate balance, referred to as self-organized criticality[12], where system state can be driven out of equilibrium. For example, a system in equilibrium may experience turbulence, where system state becomes perturbed and moves through a set of points for some period of time before returning to equilibrium. Several natural systems appear to exhibit punctuated equilibria[13], where the system state moves through occasional periods of turbulence with a frequency inversely related to magnitude. Similarly, a system in equilibrium may transition to chaos or oscillation (or perhaps move toward a less ordered state). Movements among these various patterns (see Figure 1) are usually considered to be phase transitions.[14]

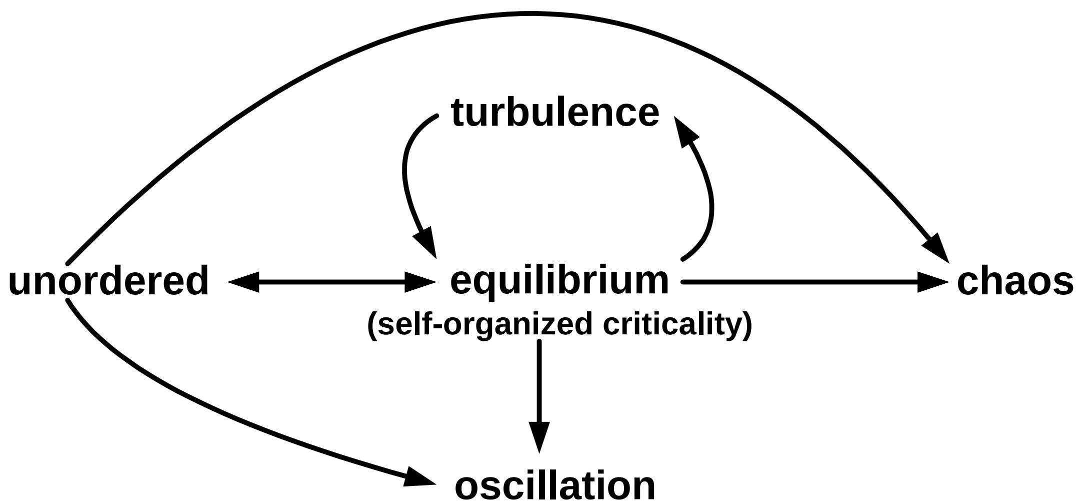

Figure 1: Possible Phase Transitions Among System States

Investigations of many natural and man-made dynamic systems reveal that phase transitions occur quickly after reaching some threshold. For example, Kuramoto shows[15] a system of coupled oscillators remains desynchronized until coupling strength reaches a critical threshold after which synchronization advances in stages that increase the coupling strength, which further increases the scope of synchronization. Floyd and Jacobson observe[9] that network traffic becomes synchronized only when the number of sources exceeds some transition threshold. Roli and Zambonelli report[16] that a dissipative cellular automata model exhibits a macroscopic spatial structure as soon as external stimulation reaches a threshold value and exhibits a chaotic pattern once external stimulation passes a higher threshold value. In a study of random graphs, Erdös and Rényi identified[17] a phase transition occurs as the number of randomly placed links exceeds half the number of nodes, after which a graph becomes fully connected.

Why do so many dynamic systems exhibit self-organizing properties? What benefits does self-organization convey? Literature review suggests five major benefits arising from self-organization. Adaptability is a key benefit both in the short run and in the long run. Short-term flexibility allows a dynamic system to maintain stable operating states in the face of changing environmental conditions.[18] Long-term evolution enables a dynamic system to develop new equilibrium states in response to shifting environmental patterns. This evolutionary response suggests a second benefit – self-organization increases the problem solving range of a system.[19] Evolution implies memory, which implies learning. Beyond that, self-organizing systems can solve problems that might be unsolvable currently using existing analytical techniques.[1] Even for currently solvable problems, self-organizing systems can devise innovative solutions that might otherwise go undiscovered.[20] Many self-organizing systems also exhibit a third benefit: short-term robustness and long-term survivability. Lacking a point of central control and possessing an ability to adapt to changing conditions, self-organizing systems can overcome the failure of individual components.[21] In the longer term, a self-organizing system can continue to pursue system-wide goals even beyond the lifetime of all current, system components.[22] Scalability is a fourth benefit from self-organization.[22] Self-organizing systems may grow without bound because complete information does not need to be disseminated throughout the system and processed by all components. Despite the potential for large scale, self-organizing systems prove quite efficient (fifth benefit) at solving difficult optimization problems. Self-organizing systems exhibit the principle of least action, which tends to minimize distance to an optimal (stable) state.[1]

Detecting or measuring the presence or degree of self-organization is still the subject of significant research. Systems may self-organize in space, in time, and in spatiotemporal combinations. Generally, self-organization should exhibit some increased correlation along a dimension of measurement – implying some sort of self-similarity. For example, self-organizing systems often organize into a hierarchy where statistical characterization of spatial organization at all layers appears quite similar.[6] Alternatively, self-organizing systems can show correlations in time, such that scaled versions of various time windows (e.g., seconds, minutes and hours) yield similar statistical characteristics.[23] Physicists often "transform the autocorrelation function into the Fourier spectrum. A power-law decay for the correlations as a function of time translates into a power-law decay of the spectrum as a function of frequency…also called 1/*f* noise."[24] Fourier transforms can also reveal the presence of oscillations by identifying specific dominant frequencies.[25] Wavelet transformations may be used to find correlations among spatial or temporal scales.[26]

Other measures of self-organization have also been proposed. For example, Oprisan[27] defines three measures, angular momentum, contrast, and correlation, to describe (changes in) the level of aggregation within a spatial extent. Other researchers[28,29] leverage thermodynamics, using decrease in entropy to indicate increased order arising from self-organization. Alternatively, some researchers[30-32] formulate measures of statistical complexity such that changes in complexity point to increasing system order.

Self-Organization as a Design Principle

Noting the pervasive presence of self-organization in natural and social systems and also respecting the potential benefits of self-organizing systems, numerous researchers have begun to investigate how models of self-organization can be applied intentionally to design large, distributed systems. This section surveys some representative models, organized loosely as biological models, social models, economic models, and other models.

Biological Models. Numerous scientists have uncovered evidence of self-organization in biological processes, inspiring computer-science researchers to investigate their application to system design. For example, during biological reproduction embryos form as a collection of homogeneous cells that develops into a complex organism with specialized functions. This process of multi-cellular *embryogenesis* uses local self-coordination to enable cells to differentiate their functions. Researchers at MIT[33] are investigating the use of such techniques to enable a substrate of homogeneous (or amorphous) computers to self-organize into differentiated structure and function to solve various problems. NASA researchers[34] are also investigating embryogenesis as a means to adapt undifferentiated processors on deep-space probes in order to permit changes in spacecraft function during missions of long duration. Nagpal[35], a researcher at Harvard, has proposed a set of primitives, based on mechanisms from embryogenesis, which engineers could use to cause homogeneous processes to self-organize into desire functionality and structure.

Other researchers aim to exploit the neural development process that allows an undifferentiated collection of neurons in many different parts of the brain to self-organize into specialized pattern recognition networks that can distinguish and classify sensory inputs. For example, Kohonen[36] has developed an algorithm for *self-organizing maps* (SOMs) that can transform a multidimensional space of inputs into a lower dimensional lattice of neurons such that topological relationships among the input space are reflected into the constructed neural network. Researchers[37] apply SOM techniques to a range of system-engineering challenges.

Other human biological functions also serve as an inspiration and design model. Hofmeyr and Forrest [38], for example, define an *artificial immune system* and describe its application to intrusion detection in computer networks. IBM has founded an entire research program, autonomic computing[39], based on modeling self-managing systems after concepts inherent in the human nervous system. *Regulatory genetic systems* in living cells have been modeled as *NK* Boolean networks[3] (of *N* logic elements each with *K* inputs and one output) or probabilistic Boolean networks[40] that self-organize into attractors comprising cyclic sequences of states. *NK* networks have been applied in a range of applications, including modeling structure dynamics in industrial networks[41]and finding a combination of values that satisfies a Boolean proposition[42].

Evolutionary processes have also inspired computer scientists to apply *natural selection* to evolve solutions to a wide range of problems[43-47] that would be difficult to solve using more conventional techniques. Computer scientists have also begun to examine whether genetic mechanisms operating at the cell level might be applied to solve computing problems. For example, scientists have discovered *RNA editing*[48], which refers to any process that produces messenger RNA molecules not specifically encoded in DNA – implying a means to inject new information into evolving cells. Perhaps RNA editing can be used to encode and evolve symbol sets through which self-organizing systems could develop vocabularies for communicating.[49]

Social Models. Recently, scientists have begun studying the organization and function of *swarms*, such as birds, insects, viruses, molds, and pedestrians. All such swarms appear to exhibit self-organization arising from the ability of swarm members to exchange information with each other, either directly or indirectly through their shared environment. Direct information exchange (e.g., through visual or auditory channels) implies a synchrony in time. For example, birds can maneuver as flocks if each bird follows three general rules[50]: (1) alignment (move toward the average heading of other birds), (2) cohesion (maneuver toward the average position of other birds) and (3) separation (avoid coming too close to other birds). Similarly, large groups of fireflies can synchronize their flashing, using visual cues and internal timing mechanisms.[51] Computer scientists have also based information dissemination models on the mechanisms through which virus epidemics flow through populations.[52]

Indirect information exchange, or *stigmergy*[53], implies that swarm members are mobile; thus, some form of information must be deposited in space to be encountered by members arriving later. For example, ants can deposit a chemical (pheromone) that can attract other ants, which strengthen the scent attracting additional ants and so on. This self-organizing behavior leads to ant trails used to retrieve food and return it to the nest. As the food supply becomes exhausted, ant visits on a trail diminish, the scent decays, and the trail is eventually abandoned. Similar behavior has been observed in slime molds[54], which normally move through dirt as individual single-celled organisms until environment conditions deteriorate. A worsening environment leads cells to emit a chemical that guides collective movement so that large mold structures emerge, presumably allowing the cells to survive until the environment improves. Computer scientists have begun to apply stigmergy in many applications[55], notably robotics[56], communications routing[57], and network design.[58]

Self-organizing swarm behaviors have also been discovered in crowds of pedestrians[59], where repulsive interactions give rise to bidirectional lanes and to synchronized alternation at bottlenecks. On grass, such pedestrian interactions lead to deformations that reinforce the behaviors. With enough use the deformations become barren paths, which can later be replaced with cement pavement, yielding an emergent sidewalk design.

Economic Models. Economies are self-organizing systems where producers and consumers interact through markets to set prices under which to exchange goods and services. While most readers probably associate economics with capitalism, researchers are investigating how to design self-organizing information systems based on numerous economic models, including self-interest[60], socialism[61], communism[62], altruism[63], game theory[64], and catallaxy[65].

Other Models. A number of other self-organizing models from physics and chemistry have been applied to design computer, communications, and information systems. Such models include electromagnetism[66] (attraction-repulsion), thermodynamics[67] (entropy reduction), molecular equilibrium[68] (minimizing energy or repulsion force), diffusion[69] (chemical gradients), and phase-transition resistance[70] (stabilizing system state far from phase transition regions).

Self-Organization in Wireless Systems

Self-organization applied to wireless networks is not a new idea. Interested readers should consult a 1986 survey by Robertazzi and Sarachik[71], which connected growing viability of wireless communications with the possibility of self-organization for topology formation, route selection, transmission scheduling, and task allocation. While many problems identified in the earlier survey continue to apply 20 years later, the nature of wireless networks has become more tangible (e.g., millions of cell-phone subscribers) and exhibits potential to reach much larger sizes (e.g., sensor networks) with more frequent mobility of infrastructural components (e.g., mobile ad hoc networks). The current survey focuses mainly on self-organization techniques applied in sensor and mobile ad hoc networks, which are not yet widely deployed.

Self-organizing mechanisms could pay dividends in almost any kind of wireless network. For example, self-organization might be applied to adapt to changing user density and traffic patterns in fixed wireless networks, where only users move. Self-organization could help reconfigure topologies as nodes move in and out of range in mobile ad hoc networks, where all nodes may move. Self-organization could form an initial topology among large numbers of wireless sensor nodes dropped across a geographic area, and then adjust the topology as sensors exhaust power and more sensors are injected. Rather than address particular network types, this paper considers application of self-organization to specific functions in wireless networks. The functions are treated in five, somewhat arbitrary, categories: (1) resource sharing (e.g., of spectrum, bandwidth, and processing capacity), (2) structure formation and maintenance (e.g., of topologies, software components, and conversational syntaxes), (3) behavior shaping (e.g., of routing, information dissemination, querying, and task or service placement), (4) resource management (e.g., to synchronize time and conserve power), and (5) resiliency (e.g., repairing faults and resisting attacks). Readers should note that sometimes a self-organizing wireless network treats several of these categories in combination in order to optimize among a variety of system traits. Often one of the traits of interest will be power conservation. For example, minimizing routing latency might be balanced against maximizing battery life.

Resource Sharing

Nodes and users in a wireless network must share a number of resources, such as electromagnetic spectrum, transmission bandwidth, and processing capacity. The task becomes difficult when the number of nodes and traffic demands are not known a priori or are not fixed. Self-organization can be used to discover initial participants and demands, to determine how best to allocate resources to satisfy an existing situation, to monitor changes, and to reallocate resources as needed. Consider some specific examples drawn from the literature.

Processing. A mobile ad hoc network (MANET) requires wireless nodes not only to act as data sources and sinks but also as relays that forward packets among neighboring nodes. Assuming nodes have finite power, tradeoffs arise among network throughput (which should be as high as possible) and node lifetime (which should be as long as possible). Complete cooperation with forwarding minimizes a node’s lifetime, while completely uncooperative behavior drives throughput to zero. Srinivasan et al.[72]

describe a game-theoretic algorithm, based on Generous Tit-For-Tat, designed to drive a system of nodes to Nash equilibrium where each node achieves the best possible tradeoff among throughput and lifetime. Assuming that each node understands its maximum forwarding rate and maintains a history of its experiences regarding the rate at which its forwarding requests are honored, a node will reject a forwarding request beyond its maximum rate (outside healthy operating bounds) or if the node is forwarding more packets than another node is forwarding for it. This later decision allows a small amount of excess forwarding – representing the generous portion of the algorithm.

Buttyán and Hubaux[73] illustrate an even simpler, counter-based mechanism that requires forwarding packets for others to gain credits to originate local packets. They formulate a rule allowing a node to maximize its own packet origination rate conditioned on forwarding packets for others. For the parameters simulated, each node achieved an optimal packet origination rate by forwarding about five packets for each local packet.

Typical energy-aware routing schemes maintain a list of possible routes and then forward packets with a uniform probability among them. Willig and colleagues[74] observe that sensor networks may contain nodes with a range of capabilities, including differences in available power, and argue that network lifetime could be increased if more capable nodes handled more of the load. To enable asymmetric load assignment, Willig et al. define an altruistic (or friendly neighbor) approach, where nodes periodically announce their capabilities, location, and address, along with a time for which a node is willing to forward packets. The assumption is that only nodes with rich power sources would announce. The cost of forwarding packets over self-declared altruistic nodes is then discounted, thus increasing the probability of relaying packets through those nodes.

Simulation results show that this altruistic approach yields significant improvement in both network lifetime and response time when compared to a typical energy-aware routing scheme.

Channel. Sohrabi and Pottie[75] consider a group of randomly deployed sensor nodes that must form into a network and organize transmissions to maximize energy life. They propose a distributed algorithm where each node cycles continuously through two phases. In the first phase, a node operates in a random-access mode, first listening for invitations from other nodes or issuing such invitations. Successful invitation handshakes lead to modifications in a time-divided schedule (used in the second phase) that allow pairs of sensors to exchange data at agreed times. In the second phase, nodes exchange information according to the agreed schedule. Simulation results show that a network of 150 sensors (each powered on only about 25% of the time) can be connected within five message times.

Kompella and Snoeren[76] present a distributed algorithm that allows individual sensors sharing a channel to independently adjust transmit power and rate to conserve energy without significantly degrading channel capacity or fairness on oversubscribed channels. Kompella and Snoeren observe that when channel load is low then messages can be sent more slowly (i.e., at lower power) without building up an excessive queue, while high load requires messages to be sent more quickly to avoid excessive queuing. They define a self-organizing approach were nodes sharing a channel snoop on transmissions and use measured transmission rates to estimate the message load at each node. Once each node has sent at least one message, then all nodes can converge to a similar estimate of the channel load and each can then independently adjust its transmission speed to ensure that all queued packets get an equal share of the channel.

Wu and Biswas[77] define a self-reorganizing slot-allocation protocol that enables clustered sensor nodes to reduce interference and conserve energy while providing reasonable latencies for monitoring applications. Initially, sensors in each cluster are assigned random transmission and reception slots and then the slot allocation schedule is adapted based on feedback derived from collisions. Here, collisions could arise when sensors in different overlapping clusters attempt to transmit at the same time. Collisions are detected at individual sensor nodes (and then transmitted to a cluster head) or are inferred by a cluster head that detects when a sensor does not use its assigned slot. Each cluster head examines collision information and adjusts slot allocations in an effort to reduce collisions. The main idea is that each cluster head will swap collision slots into free slots. When insufficient free slots exist, each cluster head has the authority to increase the duration of a transmission period to create additional slots. These independent adjustments continue with each transmission period until the cluster schedule reaches a stable state, free of collisions. Wu and Biswas show that this self-organizing approach closely matches performance of an ideal slot-allocation algorithm.

Duque et al.[78] describe an approach, based on self-organizing maps, to allocate spectrum to connections in a dynamically changing cellular network. Given a set of network measurements (e.g., cell interference and channel compatibility), Kohonen's algorithm is used to construct a mapping into equivalence classes where all radio relays in a partition have similar interference situations. Subsequently, an iterative algorithm searches for variations in channel assignments that optimize network performance for a given interference situation. The self-organized maps would be distributed to radio relays, where continuous monitoring would allow relays to switch channel assignments to match changes in the interference situation.

Ho et al.[29] describe a self-organizing algorithm that allows radio relays in a cellular network to create and dynamically adjust cell sizes to maintain maximum coverage with minimum interference. Each relay will periodically listen for neighboring relays. Hearing a new neighbor (or neighbor signoff) will stimulate a relay to conduct an expanding-ring search to calculate its distance from all reachable relays. Subsequently, the relay computes and distributes a new cell size, then waits for the next listening period. Ho and colleagues use an entropy-based complexity metric to reveal some critical characteristics about the delay between listening periods. Specifically, when the delay is too short, the network never self-organizes. Above a particular delay threshold the probability of the network self-organizing increases with the delay. Beyond a second delay threshold the network always self-organizes.

Structure Formation and Maintenance

Typically wireless networks, especially sensor networks, are deployed incrementally without central planning and must adapt to changes in node density and mobility, while simultaneously maximizing node life (i.e., minimizing power consumption) and meeting performance objectives. Designing and deploying static topologies cannot satisfy such a challenging combination of requirements. For this reason, numerous researchers investigate approaches that allow wireless nodes to self-organize into efficient clustered topologies and to maintain essential cluster properties in response to changing node populations. In special cases, networks may be formed from mobile sensor platforms, which researchers consider how best to position. Some researchers have also begun to investigate how the software architecture and structure of individual nodes can be self-configured to correspond to a node's environment. A few researchers have even considered how communicating nodes can self-organize vocabularies for conversations. Consider a variety of examples found in the literature.

Topology Formation. Sensor nodes may be deployed with significant density, which could lead to redundancy in node coverage that might be exploited to extend the overall lifetime of a sensor network. Cerpa and Estrin[79] describe one means this could be achieved through a self-organizing regime where individual sensor nodes probe their local communication environment and do not join in a multi-hop routing infrastructure until some need arises. For example, after detecting a high message-loss rate, a node could request other nodes in the area to join the network in order to relay messages. A node might also reduce its duty cycle upon detecting message losses due to collisions.

Parunak and Brueckner[80] consider server placement and selection in an ad hoc network where mobile nodes with constrained power lead to continuous topology changes. They propose a self-organizing approach, based on stimergic learning, that allows a server population to maintain the minimum necessary number of nodes at locations appropriate to serve a client population and that allows clients to learn where to direct service requests. Servers implement a reinforcement-learning algorithm where they extend their lifetime based on the number of client transactions arriving within a measurement interval. Clients share with direct neighbors a history of interactions with servers. Histories are reinforced based on positive and negative server interactions. Further, histories decay over time in order to give more weight to recent interactions.

Clients eliminate memory of any server that reaches a threshold of negative performance. Simulation results show that stimergic learning leads to significant power conservation without significantly reducing performance.

Hester et al.[81] consider a network of fixed, low-power (and low-duty) sensor nodes that must form a routing topology in order to carry messages for any useful distance. They describe self-organizing algorithms for formation and maintenance of a spanning tree, where maintenance includes adjustments for both performance and reliability. Nodes periodically send beacon messages, which other nodes may hear if those nodes are powered on, listening on the same frequency and within transmission range. A beacon message includes a node's address and information about its depth in a spanning tree. A node will attach itself to a beaconing node that has the least depth (i.e., so that the path to the root is shortest). The beaconing protocol forms an initial topology and adjusts the topology in response to node failures and arrivals.

Chan and Perrig[82] define an emergent algorithm that enables a collection of sensor nodes to form a uniformly clustered topology based on simple local actions taken at each node in response to feedback from nearby nodes. A node declares itself a cluster head when the number of members that would join solely its cluster exceeds an adaptive threshold the decays exponentially with node density. Cluster heads periodically poll cluster members to determine if cluster control might more profitably migrate elsewhere, and will prompt more appropriate nodes to form clusters as needed. Simulation results show that the algorithm forms and maintains clusters of uniform size with a standard deviation of about 23%.

Sensor Placement. Some sensors are mounted on mobile platforms, which permit the option to enhance sensor coverage after initial deployment. Given that desirable coverage depends upon situation and environment, self-organizing approaches seem necessary to enable mobile sensor nodes to reorient their positions. Some researchers investigate such problems. For example, Heo and Varshney[68] consider deployment of identical, mobile sensors, which provide a means to form a topology by self-propelled node relocations. Given an initial random topology Heo and Varshney consider a distributed self-spreading algorithm aimed to achieve a uniform topology giving maximum coverage of a region of interest in minimum time and with minimum energy consumption. The algorithm mimics molecular equilibrium, as each node finds its own lowest energy level and achieves even spacing based on computing forces of repulsion. Heo and Varshney also show how their algorithm may be adapted to achieve a clustered topology.

Wong and colleagues[66] propose a technique that allows mobile sensors to reposition themselves based on computing virtual attraction and repulsion forces exerted by other sensors and obstacles. To conserve energy, sensor movements are bounded within some limited range. The algorithm described by Wong et al. uses only local information to reposition sensors to improve coverage with minimum movement. Force between nodes is relative to distance; nodes that appear too close exert repulsion and nodes that appear too distance exert attraction. A node computes the relative influence from all surrounding forces in order to select a new position. To limit movement, nodes engage in an exponential back-off procedure to determine which node should update its position when.

Low et al.[83] consider problems arising when mobile sensors with limited sensory range are deployed sparsely relative to territory and without certain knowledge regarding location of potential targets. Under such conditions, some means must be found to direct sensor movement in order to provide adequate coverage of targets while limiting interference from an excess of sensors within the same area. Low and colleagues propose an ant-based, task-allocation scheme that enables mobile sensors to self-organize into coalitions matched to the distribution of targets across areas. Each robot measures two average delays, one for encounters with other robots and one for encounters with targets, and computes their ratio, which represents task demand as observed by the robot. Robots within the same vicinity will periodically exchange ratios, along with the number of targets currently under observation by each robot. Using this information each robot can conduct a probabilistic trial to determine its dominance over other robots. Winning such trials enhances a robot's tendency to remain in the area, while losing enhances tendency to leave. Periodically, robots conduct another probabilistic trial (which also considers distances between areas) to determine whether to leave the current area.

Software Configuration. Unlike most sensor networks, mobile ad hoc networks usually operate in a heterogeneous environment where channel conditions and protocols vary with place and time. This suggests a need for nodes to sense the environment and reconfigure platform software to match. Ribeiro-Justo et al.[84] propose a monitoring and control system modeled after processes present in biological organisms, and then apply that system to maximize quality of service given to mobile users by reconfiguring node software, including software for digital-signal processors and field-programmable gate arrays, and the architecture of software components.

Suzuki and Yamamoto[85] describe an approach, modeled after the immune system, allowing system configuration policies to be determined dynamically and continuously based on measured system conditions. Pathological system conditions (e.g., server overload) are recognized as antigens that stimulate antibodies (e.g., policies for thread management, caching, and transport protocol) based on antigen concentrations. Positive and negative reinforcement signals drive the evolution of antibody generation as system conditions vary. Simulation results show that dynamic reconfiguration provides substantially superior throughput when compared against a default, static configuration selected to match nominal operating conditions.

Vocabulary Learning. Steels[86] describes a self-organizing process through which distributed agents might develop an agreed vocabulary – mapping symbols to meanings. Here, self-organization occurs as agents iteratively exchange descriptions of such mappings and develop consensus over time, as agents note and react to the degree of communication success achieved using specific mappings. Steels shows that such a process could be used to generate vocabularies of limited scale in number of agents, symbols, and meanings. This research suggests that meanings for information exchanged among communicating agents might be established without explicit preprogramming.

Behavior Shaping

Once deployed, wireless ad hoc and sensor networks perform a range of functions some generic (e.g., routing), some application-dependent (e.g., information dissemination, querying, and search), and some resource-dependent (e.g., task assignment or service placement). The dynamic nature of wireless ad hoc and sensor networks prevent a priori design of optimal behaviors to implement such functions. For this reason, numerous researchers investigate self-organizing techniques that could enable a wireless network to shape its own behaviors based on environment and need. Selected examples follow.

Routing. Within wireless ad hoc and sensor networks nodes appear with new deployments and disappear due to mobility, power exhaustion, periods of inactivity, and vulnerability to destruction. Such dynamic behavior, coupled with desire to conserve power while limiting packet latency, presents difficult challenges for routing algorithms. To address some of these challenges, Baras and Mehta[87] investigate how ant-like agents could be used to discover and maintain routes in mobile ad hoc networks. They propose a protocol where forward "ants" seek to reach particular destinations, while backward "ants" feedback success along the paths they explore. The flow of "ants" leads to construction of routing tables within each node to encode the probability and delay associated with reaching particular destinations by forwarding packets to specific neighbors. Baras and Mehta find that their ant-like algorithm provides improved packet latency when compared with a conventional routing protocol used for mobile ad hoc networks. On the other hand, they show that their ant-like algorithm exhibits significantly higher overhead in situations where nodes have high mobility.

Servetto and Barrenechea[88] investigate how interacting particle systems (modeled as randomized walks on random graphs) might provide inspiration for efficient multi-path routing in networks with a large number of fixed sensors that power themselves off and on at random times in order to conserve power. Servetto and Barrenechea define a distributed algorithm where each node computes local parameters for a random walk such that the global network will exhibit two properties: short routes and balanced forwarding load. In computing its parameters, each node uses local information augmented only by information from one-hop neighbors and from packets transiting the node.

Tang et al.[89] consider a unique problem associated with medical sensors implanted in human subjects. In particular, since radio frequency communication produces electromagnetic fields that can be absorbed by (and heat) human tissue, they propose a thermal-aware routing protocol that avoids hot spots. Temperature is estimated for points in a grid by using a continuous-time, differential, (Pennes) bioheat equation. Packets destined for a hot spot will be buffered until estimated temperature drops, and packets that cannot be delivered within a deadline are discarded. Next routing hops for packets are selected based on temperature rather than shortest path. If a packet cannot advance to a next hop (due to temperature constraints), then the packet is returned to the previous hop, which can try another path or return the packet to its previous hop, and so on. Simulation results, which compare thermal-aware routing against shortest-path routing, show that thermal-aware routing yields a smaller maximum and average temperature increase and induces less traffic congestion. On the other hand, shortest-path routing gives lower packet latencies.

Information Dissemination. Closely related to routing protocols, information-dissemination protocols push data from sources (e.g., sensors) toward destinations for which information could be relevant. Such protocols should also conserve energy, provide low latency, and tolerate node and link failures. Designing information-dissemination regimes to satisfy these properties proves a challenging task. Following are some approaches suggested by researchers.

Intanagonwiwat et al.[90] propose a directed-diffusion protocol where information, represented as attribute-value pairs, is drawn toward consumers that express an interest. A data sink periodically sends to its neighbors a task consisting of a time-to-live, an event rate, and a list of attribute-value pairs. Nodes cache each received interest, along with one or more gradients, where each gradient defines a direction of flow and a desired event rate associated with one neighbor. Interests diffuse through a network as nodes forward received interests to neighbors. Typically, a sink will disseminate a request for events to be received at a slow rate. Subsequently, the sink may evaluate the quality and timeliness of received events and then reinforce one particular neighbor by disseminating interest in a higher event rate. The reinforcement diffuses toward the nodes providing the desired data. Simulation results, comparing with a typical flooding algorithm, show that, in a network of up to 250 sensors, directed diffusion yields lower energy use and lower delay. Directed diffusion also adapts automatically to failures in sensor nodes.

Krishnamachari and Iyengar[91] focus on feature extraction in sensor networks where some percentage of sensors may provide faulty readings. Assuming faulty sensor readings are uncorrelated while measurements of meaningful features are highly correlated, Krishnamachari and Iyengar use Bayesian analysis to demonstrate that sensors should only accept their own readings when at least half of neighboring sensors provide the same reading. Under such conditions, sensor groups can reduce fault rates by up to 95% when 10% of sensors are faulty. Further, they define a self-organizing algorithm for sensors in a feature area to form a cluster, elect a leader and construct a spanning tree from cluster members to the leader. Finally, they show that a cluster could use stepwise rectangular approximation to compress transmitted data, giving an order-of-magnitude in energy savings while also providing tight approximation of a feature region.

Wischhof et al.[92] describe an ambitious research project to develop a self-organizing traffic information system, where information about traffic conditions propagates among cars moving along a road system. Some cars are assumed to be equipped with special gear (e.g., global-positioning system, wireless radio hardware and computer connected to in-car sensors). Each equipped car conducts a repeated cycle of reception, analysis, and transmission. During reception a car receives information from any cars within radio range. Based on received information a car updates its own traffic picture during an analysis phase, and subsequently transmits its updated traffic picture to cars within range. Given that cars are moving relative to each other and that cars are moving in various directions, traffic information propagates throughout the roadway. Simulation results investigate various design parameters of such a system, for example, information-dissemination rate for particular percentages of equipped cars at specific traffic densities.

Query and Search. Closely related to information-dissemination protocols, query and search protocols allow sinks to pull data from relevant sources. Such protocols should provide the same low latency, energy efficiency and failure resilience required for information dissemination. For example, Wang et al.[93] propose that attributes should be attached to sensors and then used to cluster sensors hierarchically in order to steer queries toward sensors more likely to provide relevant data. The basic idea is to organize a location-based, logical hierarchy, where sensors with the same attribute self-organize into clusters so that one elected node (the cluster head) takes responsibility for collecting information and deciding whether to forward or drop queries. Cluster-head duties rotate in order to balance load among cluster members. Cluster-head failure can also be detected, stimulating a new election. Analysis shows that routing queries through a clustered hierarchy yields lower overhead than flooding.

Braginsky and Estrin[94] consider routing queries in sensor networks without a suitable geographical organization. Their proposed solution, called rumor routing, is to propagate queries using a random walk and to allow nodes in the network to learn routes (through discovery agents) to various events in the network, and to optimize those paths over time. Once a (random-walk) query intersects with a path to an event of interest, the random walk ceases and the query follows the previously discovered path. The protocol is designed so that both discovery agents and queries have a limited time-to-live. The number of discovery agents is also a design parameter. The goal of rumor routing is to provide a tunable (energy cost vs. discovery probability) design alternative to flooding of events or queries.

Wang et al.[67] consider a specific application where sensors are used to determine a target's location. Given an estimate of location, they wish to choose a sensor to query in order to increase estimate accuracy. They propose selecting to query the sensor with information that would yield (nearly) the largest reduction in entropy associated with the probability distribution of the target's location. Simulation results show that entropy-based, sensor selection, with its lower computational requirement, works nearly as effectively as approaches based on maximizing mutual information.

Dimakis et al.[95] consider a network composed of k sensors and n storage nodes ($k < n$) that can each store the same, fixed amount of information. They propose an approach that uses decentralized erasure codes to allow (with high probability) retrieval of k sensor readings by querying any k nodes, given that sensor readings are distributed to about $O(\ln n)$ storage nodes.

Task Assignment and Service Placement. Dynamic wireless networks may require a subset of nodes to host or provide particular services, such as translating between incompatible protocols or aggregating, caching or filtering data. Deciding which nodes should perform particular functions may require consideration of the capabilities or state of individual nodes, the network topology and variations in demand. These factors suggest the need to dynamically assign tasks, roles, or services to specific nodes and then to reassign them as conditions changes. A number of researchers investigate approaches to satisfy such needs. For example, Frank and Romer[96] define a language to specify roles that nodes might play, along with rules for deciding how to assign roles. They also provide a distributed role-assignment algorithm that nodes use to determine when to adopt specified roles. The algorithm periodically follows two steps: disseminate node property information to neighbors and evaluate rules for any required changes to node role. Each step occurs over a random jitter interval, where the interval for role evaluation exceeds the interval for property dissemination. Executing this algorithm allows roles to migrate among nodes as conditions change, including node population.

Jamjoom et al.[97] consider a self-organizing approach to service placement based on changes in demand for specific types of services. Here, each node measures local demand for services. As demand passes a rising threshold the node becomes a candidate to replicate a service; as demand falls below some descending threshold the replicated service may be destroyed. A configuration parameter (varied by service type) determines whether a replica should be placed closer to the service consumer or provider. An exponential back-off algorithm controls oscillation in replica generation by freezing a node's ability to create replicas under particular conditions.

Itao and colleagues[98] investigate biologically inspired models for autonomous components to establish cooperative relationships to provide network services. Components discover other components and exchange sets of traits, such as identity, type and capabilities. Each component maintains a relationship record for other discovered components to track the number and utility of interactions. When requested to provide a service, a component may enlist other components as needed based on their capabilities and on the strength of existing relationships. Users reward service providers based upon satisfaction received; the reward function is used to increase relationship strengths among components that cooperate to provide a service.

Resource Management

Some critical resource management operations underlie most network-wide functions in wireless ad hoc and sensor networks. This survey treats two such operations: synchronization and power conservation. Several network functions depend upon distributed nodes having a shared measure of time. For example, organizing a transmission schedule to limit interference requires that neighboring nodes have a synchronized notion of period and phase. Similarly, choosing sleep and wake periods for a node suggests need for sufficient synchrony among nodes within a network. Alternating sleep and wake periods provide one means of conserving power within a wireless network. Several other options may also be implemented to extend network lifetime. Consider some self-organizing approaches to synchronize and to conserve energy.

Synchronization. Werner-Allen and colleagues[99] describe an algorithm for time synchronization based on a mathematical model representing the method used by fireflies to synchronize spontaneously. Further, these researchers provide an analysis, simulation, and implementation of the algorithm in the context of a multi-hop sensor network with asymmetric links and message losses. Results with a 24-node test bed achieve synchronization of about 130 microseconds (50th percentile) within less than five minutes.

Elson et al.[100] describe and characterize an algorithm for phase synchronization based upon having individual nodes log times at which they receive reference broadcasts and then exchange logs with neighboring nodes. Logged times can be used to perform a least-squares linear regression between pairs of receivers, which allows a timestamp from a remote node to be converted locally into an equivalent timestamp. Experiments with the proposed technique demonstrated that two small sensors could maintain synchronization within 11 microseconds on a shared channel where a third node provided reference broadcasts. Additional experiments showed that the algorithm could be used in multi-hop networks; however, precision decays – the average error is proportional to the square root of the number of hops.

Power Conservation. Most designs for wireless sensor networks consider techniques to reduce energy consumption. Two fundamental techniques include powering off radios and limiting transmission power. A number of researchers investigate how to apply these techniques without centralized control. For example, Chen and colleagues[61] observe that all nodes need not be powered on at all times in networks with sufficient density – in fact they argue that powering on too many nodes can create interference and diminish network capacity. They define a decentralized algorithm allowing nodes to make local decisions about when to sleep and when to wake up and become a forwarding node. Whenever a node discovers two neighbors cannot communicate, the node delays before volunteering to forward packets. Nodes with more power delay for a shorter time, as do nodes that would connect more neighbors. This allows nodes with best ability and greatest utility to power on, allowing less capable and beneficial nodes to remain dormant.

Conner et al.[101] investigate two complementary algorithms to increase the lifespan of sensor networks. One algorithm systematically adjusts a network topology to shift forwarding burden to energy-rich nodes, while the other algorithm enables non-forwarding nodes to sleep most of the time without missing packets. The topology-control algorithm, which adjusts based on periodic probing, favors selecting fewer forwarding nodes that are more richly connected, leading to a shallow network where most nodes can be reached within a hop or two. The node-scheduling algorithm allows a node at power up to discover (via snooping) the current schedule during which other nodes send short messages indicating any intention to send a data packet. The new node can then select an open spot in the schedule. To send a data message, a node first announces an intention to send at a particular time (avoiding known conflicts) to a particular destination, which will then know when to wake up to receive the transmission. This algorithm assumes that data transmissions will be relatively rare and that power savings may be traded for higher latency.

Xu and colleagues[102] compare two distributed approaches to select redundant nodes in a sensor network in order to turn off their radios. One approach, called geographic-adaptive fidelity, partitions a network into a virtual grid based on maximum radio range, where any node within a grid can communicate with all nodes in all adjacent grids. Nodes periodically exchange grid identifiers. Only one node within each grid remains powered on after each such exchange. The second approach, called cluster-based energy conservation, requires nodes to self-organize into clusters, where all nodes can be reached within at most two hops, and then to elect a cluster head (the most energy-rich node that can reach all cluster nodes within one hop) and gateways (i.e., nodes that can hear cluster heads or gateway nodes within other clusters). When multiple gateways exist within a cluster one is elected to remain on by giving priority to gateways that can reach other cluster heads and to gateways with more available energy. Redundant nodes in a cluster are powered off but intermittently awaken to rerun the algorithm in order to adapt to any changes.

Kubisch et al.[103] compare two node-local algorithms for adapting transmission power within fixed, wireless sensor networks. One algorithm requires nodes to periodically broadcast probe packets and to listen for acknowledgments from neighboring nodes. Failure to receive a sufficient number of acknowledgments stimulates a node to increase transmit power and retry. Receiving too many acknowledgments causes a node to decrease transmit power and retry. Receiving a target number of acknowledgments terminates a probe period and establishes a level for transmission power. The second algorithm includes in each acknowledgment the number of neighbors that can be reached by the respondent. The probe issuer computes a mean number of neighbors that it should be able to reach. If the mean is too small, then transmit power is increased and another probe is sent. If the mean is too large, then transmit power is decreased and another probe issued. Simulation results find that using these self-organizing algorithms leads to network lifetimes within a lifetime or two of the global optimum that might be achieved using centralized computations.

Resiliency

Given potential for sensor networks to be deployed in critical applications, issues arise regarding resiliency in the face of failures and attacks. A small sampling of related research follows.

Failures. Gupta and Younis[104] propose a method to recover sensors from a cluster with a failed cluster head. Their method does not require network-wide re-clustering. Fault detection depends upon cluster heads periodically exchanging vectors indicating perceived status of other cluster heads. Each cluster head uses these vectors to determine a consensus view of failed cluster heads. The interval between vector exchanges expands multiplicatively over time when all cluster heads appear operational and contracts linearly during periods when some cluster heads appear suspect. Variation in the vector-exchange cycle lowers overhead for stable topologies, yet improves responsiveness during periods of instability. Fault recovery depends upon the initial technique adopted for cluster formation, where the protocol has cluster heads identify all sensors within radio range and then partition that set into primary and backup cluster members. The partitioning places sensors into the primary set based on minimizing communication cost. During recovery, sensors in multiple backup sets are reassigned to the primary set of the cluster head that offers the lowest communication cost.

Bychkovskiy and colleagues[105] consider the problem of post-deployment calibration to remove systematic bias from sensor readings in large, dense networks.

These researchers propose a two-phase algorithm. In phase one, sensors with close spatial correlation first exchange temporally correlated readings and then devise pair-wise mapping functions between statistically relevant data points. In phase two, sensors exchange matrices of mapping functions and then independently iterate over possible paths through the mapping functions to compute the mapping outcome with the highest confidence. Improving accuracy and computational efficiency of the proposed algorithm requires further study.

Attacks. Potential attacks against wireless ad hoc and sensor networks come in a variety of forms. This survey considers two attack types: injecting false sensor reports and shortening network lifetime. Ye et al.[106] investigate a statistical mechanism to detect and drop false information within a large, dense, sensor network where elected nodes aggregate and forward readings collected by nearby sensors. The mechanism requires that a data sink possess an indexed collection of keys partitioned into disjoint sets and that each sensor is randomly assigned a subset of index-key pairs from one partition. Any sensor report is forwarded along with a message hash generated based on one of the keys (key index also forwarded) within the sensor. An aggregating node forwards a sensor report along with one hash and key index in each of some number of key partitions. While flowing through the network, probability increases that a report transits a node that shares one of the keys used to generate one of the hashes. In such a case the transit node can verify that hash and could detect a forged report because a compromised node is uable to correctly forge all hashes for an aggregated report. Analysis and simulation results suggest that the proposed mechanism could drop between 80% and 90% of injected false reports within 10 forwarding hops with an overhead of only 14 bytes per sensor report. Dropping false reports early could reduce energy consumption and extend the network lifetime by a factor of two.

Yu and Liu[107] propose a self-organizing scheme that encourages nodes in mobile ad hoc networks to cooperate and simultaneously to resist attacks aimed to degrade performance and to shorten network lifetime. Assuming that node identities may not be spoofed, the scheme requires that every sent packet be acknowledged and that acknowledgments for packets ripple back along the transmission route from the destination toward the source. Forwarding packets and receiving acknowledgments cause updates to a balance sheet indicating the net difference between the utility a node contributes to each of its neighbors and the utility each neighbor contributes to the node. Nodes continue to forward packets for neighbors unless the net negative utility falls below some threshold. Route discovery is augmented to include information about the relative net utility between a node and all other nodes on particular paths. Packets will not be forwarded along routes without sufficient net utility to ensure delivery. Among the remaining routes, a packet is forwarded on the path that promises the maximum expected utility per unit of energy. Over time, cooperating nodes reinforce their net utilities and malicious nodes are shunned.

Discussion

Designing, deploying and operating large wireless ad hoc and sensor networks will be infeasible using traditional approaches. This follows from several anticipated traits: large numbers of nodes, uncertain communication environments, continuously changing node populations, vulnerability to attacks, and frequent and significant variations in user demands. Researchers are exploring a number of approaches, inspired by biological, social, economic and other models, to enable wireless networks to self-organize and adapt to perceived changes in the environment. While such approaches appear to provide the only feasible path to deal with tomorrow's wireless networks, a number of questions remain to be investigated.

Researchers identify specific combinations of issues to address. For example, how can one improve network lifetime while achieving bounded latency? As another example, how can one identify and remove bogus traffic from a network in order to reduce power consumption and improve accuracy of sensor information? As a third example, how can one encourage nodes to transmit information while simultaneously identifying and eliminating malicious nodes from a topology? As a fourth example, how can a network topology be varied over time to provide improved network lifetime and resilience? Deploying robust sensor networks will likely require simultaneous answers to these (and other) questions. Researchers have yet to experiment with self-organizing designs that can simultaneously address multiple dimensions of performance, security and robustness. One wonders how (or whether) a reasonably complete set of design objectives might be satisfied within a self-organizing framework?

Researchers also propose a range of approaches to self-organization. These cover various economic, biological, and social mechanisms, as well as models from physics and chemistry. Do some underlying principles unify all approaches to self-organization? If so, what are these principles? If not, do selected mechanisms and models work best for specific problems? If so, what are the implications of combining various mechanisms within the same system design? Will interaction effects arise? If so, how can such effects be identified and mitigated?

Phase transitions pose another area of concern. Many natural systems tend to self-organize to a critical, stable state, where equilibrium persists for some period before a transient turbulence, followed by return to a new stable state. Other possible transitions drive a system toward oscillation or chaos. Could self-organizing wireless networks exhibit similar propensity to reach a critical equilibrium? Recall that Ho et al.[79] found critical thresholds for one of their design parameters. Below a minimum parameter value their system failed to self-organize, while their system always self-organized above a maximum value. Between these two values, probability of self-organizing increased with the value of the design parameter. This shows self-organizing networks can exhibit phase transitions. Krishnamachari et al.[70] also report phase transitions in wireless networks. In one case, they identify a critical threshold of node density that leads to global connectivity. Below the threshold a network will not achieve complete connectivity, while above the threshold a network will generate interference that wastes energy. Further, they show that increasing power to generate connectivity leads to interference, which can lower probability of achieving a global network. Krishnamachari and colleagues suggest that phase-transition analysis could help to select design parameters that enable a self-organizing wireless network to reach a desirable operating point. But what about the possibility for changing conditions to disturb equilibrium and induce periods of instability, or worst oscillation or chaos? Can such conditions be forecast, analyzed and resisted? These questions remain open.

Overall, the picture appears cloudy with regard to self-organization in wireless networks. On one hand, wireless ad hoc and sensor networks composed of large numbers of elements cannot be designed, deployed and operated without the ability for components to self-organize and self-manage. On the other hand, significant questions remain about the behaviors that would be exhibited by such self-organizing systems. This suggests need for research to develop techniques and tools to measure, analyze, visualize and understand macroscopic (or global) behavior in networks. Only with such capabilities would it prove possible to assess likely system-wide behaviors arising from self-organizing designs that attempt to address multiple dimensions of network performance, security and robustness. Without an ability to understand global consequences of particular design decisions, deploying self-organizing networks could prove to be too risky.

Conclusions

This paper surveyed recent research on self-organizing techniques applied to design and control large wireless ad hoc and sensor networks. The survey divided that research into five functional categories: (1) resource sharing, (2) structure formation and maintenance, (3) behavior shaping, (4) resource management, and (5) resiliency. The paper also provided a brief outline of current scientific thinking regarding self-organization, both as a natural property of complex adaptive systems and as a design strategy to control distributed systems. The survey raised several concerns that remain to be addressed. Can large, wireless systems be designed within a self-organizing framework to meet a comprehensive set of design objectives? Can one set of self-organizing principles be defined, understood and applied to design distributed systems? Can mixed models of self-organization operate simultaneously within a single system without muting, amplifying or interfering with each other? Will self-organized distributed systems achieve stable states, or exhibit phase transitions to turbulent periods, oscillations or chaos? Can techniques be discovered to measure, analyze, visualize and understand global behavior in large distributed systems? Large wireless ad hoc and sensor networks cannot be designed without a foundation of self-organizing strategies; yet the number and significance of unanswered questions suggests that deploying such networks would prove quite risky. Given the acute future need and the current, limited, state of knowledge, researchers must be encouraged to continue investigating key theoretical questions underlying self-organization as a principle for designing complex information systems.

References

1. W.D. Hillis (1998) “Intelligence as an Emergent Behavior or, The Songs of Eden”, *Daedalus*, *Journal of the American Academy of Arts and Sciences,* winter, pp. 175-189.
2. M.B. Holbrook (2003) Adventures in Complexity: An Essay on Dynamic Open Complex Adaptive Systems, Butterfly Effects, Self-Organizing Order, Coevolution, the Ecological Perspective, Fitness Landscapes, Market Spaces, Emergent Beauty at the Edge of Chaos, and All That Jazz, Academy of Marketing Science Review.
3. S.A. Kauffman (1993) The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press.
4. R. M. May (1973) Stability and Complexity in Model Ecosystems, Princeton University Press.
5. J.E. Cohen, E. Briand, and C. M. Newman (1990) Community Food Webs: Data and Theory, Springer Berlin.
6. D. L. Turcotte (1997) Fractals and Chaos in Geology and Geophysics (2^{nd} edition), Cambridge University Press.
7. U.S. Bhalla and R. Iyengar (1999) “Emergent properties of networks of biological signaling pathways”, *Science*, 283, 381-387.
8. B.M Yerra and D.M. Levenson (2005) “The emergence of hierarchy in transportation networks”, *The Annals of Regional Science*, Springer-Verlag, vol. 39 no. 3, pp. 541-553.
9. S. Floyd and V. Jacobson (1994) “The synchronization of periodic routing messages”, *IEEE/ACM Transactions on Networking*, vol. 2 no. 2, pp. 122-136.
10. V. Plerou, et al. (2002) “Random matrix approach to cross correlations in financial data”, *Physical Review* E 65 066126.
11. C.Gershenson and F. Heylighen (2004) “How can we think the complex?” in Managing the Complex Volume One: Philosophy, Theory and Application, Kurt Richardson (editor), Image Age Publishers.
12. P. Bak (1996) How Nature Works – the Science of Self-Organized Criticality, Copernicus.
13. K. Ito (1995) “Punctuated-equilibrium model of biological evolution is also a self-organized-criticality model of earthquakes”, *Phys. Rev.* E 52, pp. 3232–3233.
14. H E. Stanley (1971) Introduction to Phase Transitions and Critical Phenomena, Oxford University Press.
15. Y. Kuramoto (1984) Chemical Oscillations, Waves, and Turbulence, Springer-Verlag.
16. A. Roli and F. Zambonelli (2002) “Emergence of Macro Spatial Structures in Dissipative Cellular Automata” in *Proceedings of Cellular Automata: 5th International Conference on Cellular Automata for Research and Industry*, Springer Berlin.
17. P. Erdös and A. Rényi (1960) “On the evolution of random graphs”, Publ. Math. Inst. Hung. Acad. Sci. 5, 17.
18. J. J. Kay (1984) Self-organization and the thermodynamics of living systems: a paradigm, PhD Thesis, University of Waterloo.

19. N.L. Johnson et al. (1998) "Symbiotic Intelligence: Self-Organizing Knowledge on Distributed Networks Driven by Human Interaction" in *Proceedings of the Sixth International Conference on Artificial Life*, MIT Press.
20. D.S. Linden and E.A. Althshuler (1999) "Evolving Wire Antennas Using Genetic Algorithms: A Review" in *Proceedings of the First NASA/DOD Workshop on Evolvable Hardware*, IEEE Computer Society.
21. D. Story (2003) "Swarm Intelligence: An Interview with Eric Bonabeau" in OpenP2P.com, O'Reilly.
22. S.K. Mostefaoui (2003) "Self-Organising Applications: A Survey" in the *Proceedings of the International Workshop on Engineering Self-Organising Applications 2003*, Springer-Verlag.
23. T. Karagiannis, M. Molle, and M. Faloutsos (2004) "Long-Range Dependence – Ten Years of Internet Traffic Modeling" in *IEEE Internet Computing*, September-October.
24. C.R. Shalizi (2006) "Power Law Distributions, 1/f Noise, Long-Memory Time Series" in Notebooks http://www.cscs.umich.edu/~crshalizi/notebooks/.
25. H. Parunak and R.S. VanderBok (1997) "Managing Emergent Behavior in Distributed Control Systems" in *Proceedings of the ISA Tech'97*, Instrument Society of America.
26. P. Abry and D. Veitch (1998) "Wavelet analysis of long-range dependent traffic", *IEEE Transactions on Information Theory*.
27. S.A. Oprisan (2001) "Theoretical approach on microscopic bases of stochastic functional self-organization: quantitative measures of the organizational degree of the environment", *Journal of Physics A, Mathematical and general*, vol. 34, no. 37.
28. S. Wolfram, (1983) "Statistical mechanics of cellular automata", *Reviews of Modern Physics* 55.
29. L. Ho, L.G. Samuel, and J.M. Pitts (2003) "Applying Emergent Self-Organizing Behavior for the Coordination of 4G Networks Using Complexity Metrics", *Bell Labs Technical Journal* 8(1), pp. 5–25, Wiley Periodicals.
30. C.R. Shalizi (2001) Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata, PhD dissertation, University of Wisconsin.
31. C.R. Shalizi and K.L. Shalizi (2004) "Quantifying self-organization with optimal predictors", Physical Review Letters, vol. 93, no. 11.
32. M. Li and P. Vitanyi (1997) An Introduction to Kolmogorov Complexity and Its Applications (2nd edition), Springer.
33. H. Abelson et al. (2000) "Amorphous Computing", *Communications of the ACM*, vol. 43, no. 5.
34. K.N. Lodding (2004) "Hitchhikers Guide to Biomorphic Software", *ACM Queue*, vol. 2, no. 4.
35. R. Nagpal (2003) "A Catalog of Biologically-Inspired Primitives for Engineering Self-Organization" in Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering, vol. 2977, Springer Berlin.
36. T. Kohonen (1989) Self-Organization and Associative Memory (3rd edition) Springer-Verlag.
37. T. Kohonen et al. (1996) "Engineering applications of the self-organizing map" in *Proceedings of the IEEE*, vol. 84 no. 10.

38. S. Hofmeyr and S. Forrest (2000) “Architecture for an Artificial Immune System”, *Evolutionary Computation*, vol. 7 no. 1, Morgan-Kaufmann.
39. J.J. Ritsko et al. (2003) *IBM Systems Journal*, special issue on Autonomic Computing, vol. 42, no. 1.
40. I. Shmulevich, E.R. Dougherty and W. Zhang (2002) “From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks” in Proceedings Of The IEEE, vol. 90, no. 11.
41. I.F. Wilkinson and J.B. Wiley (2000) “Modeling the Structural Dynamics of Industrial Networks” in *Proceedings of the 4th International Conference on Complex Systems* (ICCS).
42. M.Milano and A. Roli (2000) “Solving the Satisfiability Problem through Boolean Networks”, Lecture notes in Artificial Intelligence, vol. 1792.
43. J.R. Koza (1992) Genetic Programming – On the Programming of Computers, MIT Press.
44. J.R. Koza (1994) Genetic Programming II – Automatic Discovery of Reusable Programs, MIT Press.
45. Z. Michalewicz (1992) Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag.
46. D.B. Fogel (1995) Evolutionary Computation – Toward a New Philosophy of Machine Intelligence, IEEE Press.
47. D.B. Fogel, editor (1998) Evolutionary Computation – The Fossil Record, IEEE Press.
48. Arts, G.J. and R. Benne (1996) “Mechanism and evolution of RNA editing in kinetoplastida”, *Biochimica et Biophysica Acta*, vol. 1307.
49. L. M. Rocha (2000) “Syntactic autonomy, cellular automata, and RNA editing: or why self-organization needs symbols to evolve and how it might evolve them” in Closure: Emergent Organizations and Their Dynamics, Annals of the New York Academy of Sciences. Vol. 901.
50. C. W. Reynolds (1987) “Flocks, Herds, and Schools: A Distributed Behavioral Model”, *Computer Graphics*, vol. 21, no.4.
51. A. Moiseff and J. Copeland (1995) “Mechanisms of synchrony in the North American firefly *Photinus carolinus* (*Coleoptera: Lampyridae*)”, *Journal of Insect Behavior* 8.
52. D. Agrawal, A. El Abbadi, and R.C. Steinke (1997) “Epidemic Algorithms in Replicated Databases”, *Proceeding of the 16th Symposium on Principles of Database Systems*.
53. P. P.Grassé (1959). “La reconstruction du nid et les coordinations inter-individuelles chez Bellicositermes natalensis et Cubitermes sp. La theorie de la stigmergie: Essai d'interpretation des termites constructeurs”, Ins. Soc.
54. J.T. Bonner and D.S. Lamont (2005) “Behavior of cellular slime molds in the soil”, *Mycologia*, vol. 97 no. 1.
55. M. Dorigo et al., editors (2004) Ant Colony Optimization and Swarm Intelligence, 4th International Workshop, Springer-Verlag.
56. R. Beckers, O.E. Holland and J. Deneubourg (1994) “From Local Actions to Global Tasks: Stigmergy in Collective Robotics”, Artificial Life IV, MIT Press.

57. H. Yun, A.N. Zincir-Heywood (2004) "Intelligent Ants for Adaptive Network Routing", Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04).
58. M. Qoubutra, P. Premprayoon and P. Wardkein (2005) "Topological Communication Network Design using Improved Ant Colony Optimization", *Proceedings of Network and Communication Systems*.
59. D. Helbing et al. (2001) "Self-organizing pedestrian movement", *Environment and Planning B: Planning and Design*, vol. 28 no. 3.
60. D. Irwin, L. Grit and J. Chase (2004) "*Balancing Risk and Reward in a Market-based Task Service*", Proceedings of the 13th Symposium on High Performance Distributed Computing.
61. B. Chen et al. (2002) "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", *ACM Wireless Networks Journal*, 8(5), pp. 85-96.
62. A. Vahdat et al. (2002) "Self-Organizing Subsets: From Each According to His Abilities, To Each According to His Needs", Proceedings *of the 1st International Workshop on Peer-to-Peer Systems* (IPTPS '02).
63. V. Hilaire et al. (2004) "A Formal Approach to Design and Reuse Agent and Multiagent Models" in Agent-Oriented Software Engineering V, Springer.
64. R. Mahajan et al. (2004) "Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems", *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*.
65. T. Eymann, B. Padovan, D. Schoder (2000) "The Catallaxy as a new Paradigm for the Design of Information Systems", *Proceedings of the 16th IFIP World Computer Congress, Conference on Intelligent Information Processing*.
66. T. Wong, T. Tsuchiya and T. Kikuno (2004) "A Self-organizing Technique for Sensor Placement in Wireless Micro-Sensor Networks", *Proceedings of the 18th International Conference on Advanced Information Networking and Application,* IEEE, pp. 78-83.
67. H. Wang et al. (2004) "Entropy-based Sensor Selection Heuristic for Target Selection", *Proceedings of Information Processing in Sensor Networks*, ACM.
68. N. Heo and P. Varshney (2003) "An Intelligent Deployment and Clustering Algorithm for a Distributed Mobile Sensor Network", *Proceedings of the IEEE International Conference On Systems Man And Cybernetics*, pp. 4576-4581.
69. J. Faruque, K. Psounis and A. Helmy (2005) "Analysis of Gradient-Based Routing Protocols in Sensor Networks", *Distributed Computing in Sensor Systems: First IEEE International Conference*, Springer Berlin.
70. B. Krishnamachari et al. (2003) "On the Complexity of Distributed Self-Configuration in Wireless Networks", *Telecommunication Systems* 22:1–4, Kluwer Academic Publishers, pp. 33–59.
71. T.G. Robertazzi and P.E. Sarachik (1986) "Self-Organizing Communication Networks", *IEEE Communications*, vol. 21 no. 1.
72. V. Srinivasan et al. (2003) "Cooperation in Wireless Ad Hoc Networks", *Proceedings of IEEE INFOCOM*.

73. L. Buttyán and J-P Hubaux (2003) "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *Mobile Networks and Applications*, 8, pp. 579–592, Kluwer Academic Publishers.
74. A. Willig et al. (2002) "Altruists in the PicoRadio Sensor Network", *Proceedings of IEEE Workshop on Factory Communication Systems*.
75. K. Sohrabi and G.Pottie (1999) "Performance Of A Novel Self-Organization Protocol For Wireless Ad-Hoc Sensor Networks", *Proceedings of the IEEE 50th Vehicular Technology Conference*.
76. R. Kompella and A. Snoeren (2003) "Practical Lazy Scheduling in Sensor Networks", *Proceedings of SenSys'03*, ACM, pp. 280-291.
77. T. Wu and S. Biswas (2005) "Reducing Inter-Cluster TDMA Interference by Adaptive MAC Allocation in Sensor Networks", *Proceedings of the 1st International IEEE WoWMoM Workshop on Autonomic Communications and Computing* (ACC'05), pp. 507-511.
78. M Duque-Anton, B Ruber, and U Killat (1997) "Extending Kohonen's Self-Organizing Mapping for Adaptive Resource Management in Cellular Radio Networks", *IEEE Transactions on Vehicular Technology*, August, pp. 560-568.
79. A. Cerpa and D. Estrin (2002) "ASCENT: Adaptive Self-Configuring sEnsor Networks Topologies," *Proceedings of IEEE INFOCOM*, pp. 1278-1287.
80. H. Parunak and S. Brueckner (2004) "Stigmergic Learning for Self-Organizing Mobile Ad-Hoc Networks (MANET's)", *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1324-1325.
81. L. Hester et al. (2002) "neuRFon Netform: A Self-Organizing Wireless Sensor Network," http://citeseer.nj.nec.com/572360.html
82. H. Chan and A. Perrig (2004) "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", *Proceedings of the 1st European Workshop on Wireless Sensor Networks*, Vol. 2920, Springer Berlin, pp. 154 – 171.
83. K. H. Low et al. (2004) "Task Allocation via Self-Organizing Swarm Coalitions in Distributed Mobile Sensor Network", *Proceedings of the 19th National Conference on Artificial Intelligence*, AAAI, pp. 28-33.
84. G. Ribeiro-Justo et al. (2001) "Intelligent Decision-making within 4th Generation Wireless Networks", *Proceedings of the 2001 International Conference on Internet Computing*.
85. J. Suzuki and Y. Yamamoto (2000) "Building an Artificial Immune Network for Decentralized Policy Negotiation", *Proceedings of the 4th World Multiconferences on Systemics, Cybernetics, and Informatics*.
86. L. Steels (1996) "Self-organising vocabularies", *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, pp. 179-184.
87. J. Baras and H. Mehta (2003) "A Probabilistic Emergent Routing Algorithm for Mobile Ad Hoc Networks", *Proceedings of WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*.
88. S. Servetto and G. Barrenechea (2002) "Constrained Random Walks on Random Graphs: Routing Algorithms for Large Scale Wireless Sensor Networks", *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 12-21.

89. Q. Tang et al. (2005) "TARA: Thermal-Aware Routing Algorithm for Implanted Sensor Networks", *Proceedings of the International Conference on Distributed Computing in Sensor Systems* (DCOSS).
90. C. Intanagonwiwat et al. (2000) "Directed diffusion: a scalable and robust communication paradigm for sensor networks", *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, ACM Press, pp. 56-67.
91. B. Krishnamachari and S. Iyengar (2003) "Efficient and Fault-Tolerant Feature Extraction in Wireless Sensor Networks", *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, Springer-Verlag, pp. 488–501.
92. L. Wischhof et al. (2003) "SOTIS – A Self-Organizing Traffic Information System", *Proceedings of the 57th IEEE Vehicular Technology Conference*.
93. K. Wang et al. (2005) "Attribute-Based Clustering for Information Dissemination in Wireless Sensor Networks", *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*.
94. D. Braginsky and D. Estrin (2002) "Rumor Routing Algorithm For Sensor Networks", *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks*, pp. 22-31.
95. A. Dimakis et al. (2005) "Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes", *Proceedings of the Symposium on Information Processing in Sensor Networks*, pp. 111-117.
96. C. Frank and K. Romer (2005) "Algorithms for Generic Role Assignment in Wireless Sensor Networks", *Proceedings of ACM SenSys '05*.
97. H. Jamjoom et al. (1999) Self-Organizing Network Services, University of Michigan Technical Report CSE-TR-407-99.
98. T. Itao et al. (2002) "Service Emergence based on Relationship among Self-Organizing Entities", *Proceedings of the 2002 Symposium on Applications and the Internet*, IEEE.
99. G. Werner-Allen et al. (2005) "Firefly-Inspired Sensor Network Synchronicity with Realistic Radio Effects", *Proceedings of ACM SenSys '05*.
100. J. Elson et al. (2002) "Fine-Grained Network Time Synchronization using Reference Broadcasts", *Proceedings of the 5th Symposium on Operating Systems Design and Implementation*, pp. 147-163.
101. W. S. Conner et al. (2003) "Experimental Evaluation of Synchronization and Topology Control for In-Building Sensor Network Applications", *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp. 38-49.
102. Y. Xu et al. (2003) Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks, Technical Report 6, Center for Embedded Networked Computing, University of California, Los Angeles.
103. M. Kubisch et al. (2003) "Distributed Algorithms for Transmission Power Control in Wireless Sensor Networks", *Proceedings of the IEEE Wireless Communications and Networking Conference*.

104. G. Gupta and M. Younis (2003) “Fault-Tolerant Clustering of Wireless Sensor Networks”, *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 1579-1584.
105. V. Bychkovskiy et al. (2003) “A Collaborative Approach to In-Place Sensor Calibration”, *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*.
106. F. Ye et al. (2004) “Statistical En-route Filtering of Injected False Data in Sensor Networks”, *Proceedings of IEEE INFOCOM*.
107. W. Yu and K. Liu (2005) “Stimulating Cooperation and Defending Against Attacks in Self-Organized Mobile Ad Hoc Networks”, *Proceedings of the 2nd Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks,* pp. 65-75.